

# CESNET Technical Report 12/2009

## iHDTV Protocol Implementation for UltraGrid

MILOŠ LIŠKA, MARTIN BENEŠ, PETR HOLUB

Received 11.12.2009

### Abstract

This report describes implementation of iHDTV video conferencing protocol for UltraGrid. In addition to the compatibility with the original iHDTV tool implementation of this protocol allows for splitting of the video stream and sending it through two different network interfaces. This allows to send a stream of uncompressed HDTV video, which requires 1.2Gbps or 1.5Gbps of available bandwidth, over a GE network infrastructure.

*Keywords:* UltraGrid, iHDTV, iHDTV protocol, iHDTV compatibility layer

## 1 Introduction

An uncompressed HDTV video transmissions provide both high image quality and low end-to-end latency creating an unparalleled experience for the users. Such a transmissions may be successfully utilised in advanced collaborative environments, high-definition videoconferencing, distribution of remote high-definition visualisations, applications in medical area etc. In recent years transmissions of uncompressed HDTV video have been implemented by several teams. The original version of UltraGrid system was provided by Perkins and Gharai [4]. This version of UltraGrid is however discontinued now. The CESNET team has modified the original UltraGrid version [3], added a number of new features and continues with its development. There is also an iHDTV system<sup>1</sup> developed at the University of Washington in collaboration with the ResearchChannel consortium<sup>2</sup> and the HDTV over IP system iVisto by NTT laboratories [2] available for the transmissions of uncompressed HDTV video.

The number of existing implementations, which all rely on their own video formats and methods of transmission of the video stream over the IP network, complicates the deployment of systems utilising uncompressed HDTV transmissions. All these implementations usually rely on state of the art equipment (especially video acquisition or display cards), demand high computing performance and are themselves quite difficult to deploy. Thus, the users who already adopt one of the uncompressed HDTV transmissions systems usually tend to stick with it and are not really enthusiastic about adopting another one. Out of the above mentioned uncompressed HDTV video transmission systems is iHDTV one of the most widespread and used one. To enable interoperability between users of the iHDTV and UltraGrid systems and allow them to build shared collaborative environments

---

<sup>1</sup> <http://www.washington.edu/ihdtv/>

<sup>2</sup> <http://ihdtv.sourceforge.net/>

and applications, we have decided to implement an iHDTV compatibility mode into our UltraGrid version. The compatibility between iHDTV and UltraGrid should also allow for more widespread adoption of the uncompressed HDTV transmissions systems.

## 2 iHDTV

iHDTV is a videoconferencing system supporting real-time transmissions of full 1080i HDTV video along with 6 channels of audio. Transmission of the video generates about 1.5 Gbps of traffic. The main advantage of iHDTV in comparison to other systems is that iHDTV allows to split the stream into two and send it over two distinct network interfaces<sup>3</sup>. Thus it is possible to use lower-cost 1 GE infrastructure where 10 GE infrastructure would be otherwise needed to transmit such a stream.

The iHDTV was developed primarily on MS Windows XP platform. A limited subset of the iHDTV functionality is also available on Linux. So far iHDTV supports Blackmagic Design<sup>5</sup> Decklink and Intensity (Pro) cards for video acquisition and AJA Xena<sup>6</sup> cards for video displaying. Both options are supported only on MS Windows through DirectX Directshow<sup>7</sup>. iHDTV on the Linux platform than supports only rudimentary file access to video dumps and displaying of the video using XVideo extension and regular graphics cards.

## 3 Video Protocol Specification

iHDTV implements transmissions of video encoded using v210 10 bit 4:2:2 YUV pixel format [1]. The video frame rate is 29.97 frames per second while progressive frames are used. Each frame consists of 1920×1080 pixels encoded in v210 format which results in 5,529,600 bytes per frame.

Each frame is divided into top and bottom half, which are processed separately and can be sent over different network interfaces. Both halves of the frame are split and packed into UDP packets of maximum size of 8,192 B. This size was chosen, because it is possible to send such packets through most of the common switches and routers, while keeping the packets number relatively low, which reduces the network overhead. Using 8,192 B large packets allows to embed 8,128 B of iHDTV data payload, along with necessary network headers like IPv4/IPv6 header, Ethernet header, optional 802.1q header and UDP header.

The iHDTV data payload is packed into the UDP packet together with an auxiliary header. The structure of this header is illustrated in Table 1. *Stream* ID is set to

<sup>3</sup> From the networking point of view if both network interfaces were to use different IP addresses in the same subnet, then both ports would also use the same default gateway address, and the system would attempt to send all 1.5 Gbps traffic out through just one interface. Therefore, for each of the two 1 GE network interfaces on each system, separate subnets with different gateway addresses are needed. For more details on this issue see the “Functional description” page in the iHDTV wiki<sup>4</sup>.

<sup>5</sup> <http://www.blackmagic-design.com/>

<sup>6</sup> <http://www.aja.com/products/xena/>

<sup>7</sup> <http://msdn.microsoft.com/en-us/directx/default.aspx>

**Table 1.** iHDTV data payload header.

	Stream ID	Offset	Frame Number	Data Payload (Video/Audio)
Size [B]	4	4	8	8,112 (maximum)

0 for the top half of a frame and to 1 for the bottom half. Other values are used for audio, which is not covered by this document. *Offset* defines offset of video data into half-frame. It is measured in bytes. *Frame Number* defines the number of a frame, the packet payload belongs to, and is incremented with each new frame. In the case when iHDTV uses two different interfaces, frame numbers of each half-frame are dealt separately and usually differs for each half of a frame. Finally, *Data* is the actual video or audio data transferred. It can be smaller than 8,112 B in the case of the last packet of a half-frame.

## 4 Protocol Implementation in UltraGrid

UltraGrid itself relies on Real-time Transport Protocol (RTP) implementation used for delivery of video and audio streams. The RTP implementation in UltraGrid is built on UDP. Fortunately, UltraGrid code base is very modular. Thus, it turned out to be quite straightforward to reuse the existing UDP implementation and add iHDTV protocol implementation. On the other hand, the iHDTV documentation is insufficient and its code extremely difficult to understand. Therefore most of the work was to decode the iHDTV data stream, which we have done partly by analysis of the source codes and partly by analysing the streams transmitted by iHDTV using network sniffers like Wireshark<sup>8</sup>.

### 4.1 Sender

Video frames acquired from capture card are processed in a single separated POSIX thread. The only major difference in comparison to the previous video frames handling in UltraGrid is that the individual video frames are being split into the top and bottom halves and sent as two separate streams. Each half of the frame is being packed into 8,112 bytes long iHDTV packets, which are immediately packed into UDP packets and sent using standard UNIX system calls over a different datagram socket. The user can decide to specify only a single target IP address to address the remote iHDTV system. In such case the video can be transmitted only using a local 10 GE network interface and a 10 GE infrastructure. The user can also specify two IP addresses of the remote iHDTV system in order to split sending of both video streams to two 1 GE network interfaces. In such case it is again necessary to configure separate subnets with different gateway addresses for both network interfaces.

Concerning the video acquisition and sending in UltraGrid to be received by remote iHDTV system, it is important to consider that iHDTV expects the video encoded using the v210 pixel format. To this end UltraGrid supports acquisition of video encoded in v210 pixel format only on the MacOS X platform where v210

<sup>8</sup> <http://www.wireshark.org/>

is a default pixel format for handling of 10 bit 4:2:2 YUV video.

## 4.2 Receiver

Receiving iHDTV data is also done in a single separated thread. UltraGrid in the iHDTV compatibility mode is listening on a pair of local sockets for the video data. The received top and bottom halves of individual frames are then composed into a single video frame which is then displayed. The video frame is considered complete once enough video packets is received and processed (682 packets per frame by default).

This straightforward receiving scheme however appears to be insufficient. The reason is, that in the case when iHDTV sends data over two different network interfaces at the same time, there is no relation between “frame numbers” in the packets being received for the top and the bottom half of a frame.<sup>9</sup> The receiver may thus for example receive packets with frame number 356 belonging to the top half of a frame, while at the same time it receives packets with frame number 411 belonging to the bottom half of a frame. This makes it difficult to synchronize both halves of a video frame while also being able to effectively handle lost packets. In case that some packets get lost during the transfer, incomplete video frame is sent to display once the first packet from subsequent video frame is received. We opt for this behavior in order to avoid buffering and possible issues while trying to synchronize displaying of different halves of different frames.

The iHDTV compatibility mode was and released in the 20090105 development release of UltraGrid<sup>10</sup>. Originally, displaying the video sent by iHDTV required MacOS X Quicktime together with AJA Kona 3 display method because of its inherent support of the v210 pixel format.

Recent development of the UltraGrid platform featured major overhaul of the original displaying methods. The AJA Kona 3 display method was generalized to implement displaying of the video using Quicktime Video Display Component and any HW displaying card supporting this interface (e.g., AJA Kona 3<sup>11</sup>, Blackmagic Design Multibrige and others). Currently it is also possible to utilise the software-based SDL<sup>12</sup> display method and traditional graphics card to display the video with v210 pixel format as we have added transcoding of various pixel formats to 8 bit 4:2:2 YUV to the SDL display module in UltraGrid. Also, the original video displaying method using AJA Kona 3 card for the iHDTV compatibility mode was MacOS X dependent feature while the SW based SDL displaying module is available on both Linux and MacOS X platforms. Note that these revised displaying

<sup>9</sup> This behavior in iHDTV is strange. If iHDTV is sending both halves of one frame over a single interface than all packets sent for both halves of the frame in question have the same frame number. We have implemented the iHDTV compatibility layer in UltraGrid so that the frame numbers are always corresponding for both halves of the frame no matter if being sent over a single network interface or two network interfaces. We have empirically verified that in such case iHDTV is able to compose the frame from both halves with no issues.

<sup>10</sup> <http://www.sitola.cz/files/ultragrid-20090105.tar.gz>

<sup>11</sup> <http://www.aja.com/products/kona/>

<sup>12</sup> <http://www.libsdl.org/>

methods are available in the recent nightly builds of the UltraGrid provided at the home page<sup>13</sup>.

### 4.3 Other Considerations

Our current implementation of iHDTV compatibility mode in UltraGrid does not consider sending and receiving of audio data. Handling of the audio data relies on the Stream ID field in the iHDTV packet header. Thus it is very simple to drop any packets where the Stream ID does not correspond to the top or bottom half of video frame. On the sending side UltraGrid generates only those packets carrying again either top or bottom halves of the video frames. We have empirically verified that iHDTV does not insist on receiving the packets identified by Stream ID to contain audio data.

## 5 Using UltraGrid in the iHDTV Compatibility Mode

Use of the iHDTV protocol in UltraGrid is invoked by the `-i` command line switch. So for example to send the video using UltraGrid using single 10GE link, one would call the following command on MacOS X machine:

```
uv -t quicktime -g gui -i <target ip>
```

Sending the video over two 1GE network interfaces:

```
uv -t quicktime -g gui -i <target ip 1> <target ip 2>
```

Receiving the video over a single 10GE network interface and displaying it using the SDL display module:

```
uv -d sdl -g 1920:1080:v210 -i <source ip>
```

Receiving the video transmitted over a single 10GE network link and displaying it using AJA Kona 3 card on MacOS X<sup>14</sup>:

```
uv -d kona -g <device>:<mode>:v210 -i <source ip>
```

If the user does not care about the address we receive the video data from, we can just omit the source address and display the video as in the previous case<sup>15</sup>:

```
uv -d kona -g <device>:<mode>:v210 -i
```

<sup>13</sup> <http://ultragrid.sitola.cz/>

<sup>14</sup> The device and mode numbers depend on actually used Video Display Component HW and its drivers.

<sup>15</sup> This is potentially dangerous. If two distinct iHDTV senders would be sending the video streams to a receiver with no source address specified, the receiver would try to process all incoming packets and thus display garbled image.

## 6 Conclusions and Future Work

In this technical report we have described the implementation of the iHDTV compatibility mode for low-latency uncompressed HDTV video transmission system UltraGrid. We have successfully implemented and tested sending and receiving of video using the iHDTV protocol. This achievement becomes of particular importance because the NTT laboratories iVisto system also recently implemented an iHDTV compatibility layer. Thus all three competing systems (UltraGrid, iHDTV and iVisto) are able to cooperate using a common stream transmission format.

As for the future work, there remains to implement transmission of the audio data compatible with the iHDTV system. Also, receiving of the video being sent by iHDTV over two network interfaces remains a problem to be properly solved in UltraGrid.

## 7 Acknowledgements

Our work on the UltraGrid system is supported by the research intent MSM6383917201.

## References

- [1] HODDIE, P.; CHERNA, T.; PIRAZZI, C. Uncompressed Y'CbCr Video in QuickTime Files. *Developer Connection*, Apple, Inc., 14 Dec 1999 [cit. 2009-12-11]. Available online<sup>16</sup>.
- [2] HARADA, K.; KAWANO, T.; ZAIMA, K.; HATTA, S.; MENO, S. Uncompressed HDTV over IP Transmission System using Ultra-high-speed IP Streaming Technology. *NTT Tech. Rev.* 2003, vol. 1, no. 1, p. 84–89.
- [3] HOLUB, P.; MATYSKA, L.; LIŠKA, M.; HEJTMÁNEK, L.; DENEMARK, J.; REBOK, T.; HUTANU, A.; PARUCHURI, R.; RADIL, J.; HLADKÁ, E. High-definition multimedia for multiparty low-latency interactive communication. *Future Generation Computer Systems.* 2006, vol. 22, no. 8, p. 856–861.
- [4] PERKINS, C.; GHARAI, L.; LEHMAN, T.; MANKIN, A. Experiments with Delivery of HDTV over IP Networks. In *Proceedings of the 12th International Packet Video Workshop*, Pittsburgh, PA, USA, 2002. Available online<sup>17</sup>.

<sup>16</sup> <http://developer.apple.com/quicktime/icefloe/dispatch019.html>

<sup>17</sup> <http://ultragrid.east.isi.edu/publications/pv2002.pdf>